# 基于加权 K 近邻的改进密度峰值聚类算法 *

杨　震，王红军

(国防科技大学，合肥 230037)

**摘　要：**密度峰值聚类算法是一种新颖的密度聚类算法，但是，原算法仅仅考虑了数据的全局结构，在对分布不均匀的数据集进行聚类时效果不理想，并且原算法仅仅依据决策图上各点的分布情况来选取聚类中心，缺乏可靠的选取标准。针对上述问题，提出了一种基于加权 K 近邻的改进密度峰值聚类算法，将最近邻算法的思想引入密度峰值聚类算法，重新定义并计算了各数据点的局部密度，并通过权值斜率变化趋势来判别聚类中心临界点。通过在人工数据集上与 UCI 真实数据集上的实验，将该改进算法与原密度峰值聚类算法、K-MEANS 算法及 DBSCAN 算法进行了对比，证明了改进算法能够在密度不均匀数据集上有效完成聚类，能够发现任意形状簇，且在三个聚类性能指标上普遍高于另外三种算法。

**关键词：**数据挖掘；加权 K 近邻；密度峰值；聚类

**中图分类号：**TP301.6　　doi: 10.19734/j.issn.1001-3695.2018.08.0656

## Improved density peak clustering algorithm based on weighted K-nearest neighbor

Yang Zhen, Wang Hongjun

(*National University of Defense Technology, Hefei* 230037, *China*)

**Abstract:** The density peak clustering algorithm was a new density-based clustering algorithm, the algorithm requires only one input parameter and does not require frequent iterative processes. However, the original algorithm only considers the global structure of the data, and the effect is not ideal when clustering data sets with uneven distribution. Moreover, the original algorithm only selects the cluster center according to the distribution of points on the decision graph, which is not reliable. Aiming at the above problems, an improved density peak clustering algorithm based on weighted K-nearest neighbor is proposed. The idea of nearest neighbor algorithm is introduced into the density peak clustering algorithm, the local density of each data point is redefined and calculated, and determine the critical point of the cluster center by the trend of the slope of the weight. The improved algorithm is compared with the original density peak clustering algorithm, K-MEANS algorithm and DBSCAN algorithm by experiments on the artificial dataset and UCI real dataset. It is proved that the improved algorithm can deal with the density uneven dataset and find clusters of arbitrary shapes. On the three cluster performance indicators, the improved algorithm is generally higher than the other three algorithms.

**Key words:** data mining; weighted K-nearest neighbor; density peaks; clustering



## 0　引言

聚类通常作为一种无监督学习方法被用于数据挖掘领域。聚类分析的主要目的是将给定的集群划分为具有共同特征的群组，特征相似的对象被分在一起，而特征差异较大的对象则属于不同的群组。聚类在探索性模式分析，分组决策和机器学习情境中用途广泛，包括文档检索，图像分割和模式分类等[1]。聚类方法按照原理不同一般可分为五类[2]：划分聚类（如 K-means++[3]）、层次聚类、密度聚类（如 DBSCAN[4,5]）、网格聚类以及模型聚类，每种方法都有对应的优点和缺点。

其中，基于密度的聚类算法假设聚类结构能够通过样本分布的紧密程度确定，其优势在于可发现任意形状的簇。Rodriguez 等人提出了一种新颖的密度聚类算法：密度峰值聚类（density peaks clustering, DPC）算法[6]，该算法能够检测非球形簇，并且不需要用户先验指定聚类数量。DPC 算法只有截断距离 $d_c$ 这一个输入参数，因此稳定性良好。目前，已经有不少学者将这种方法用于图像处理、模式分类等领域[7-12]。

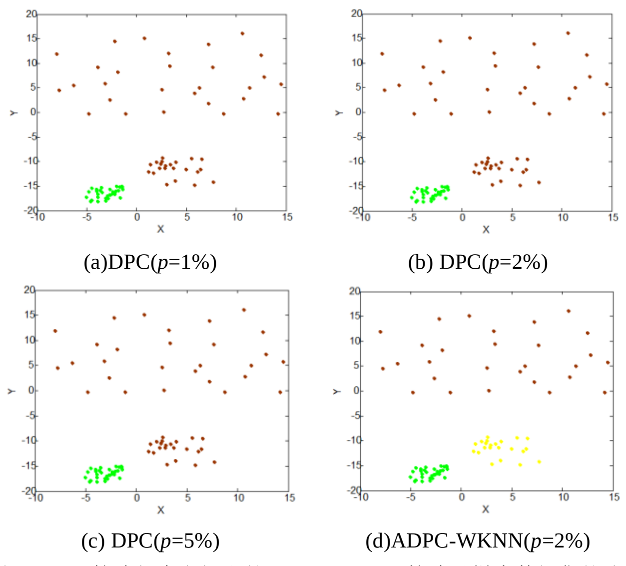
但是，DPC 算法仍然存在一些不足：a)在计算局部密度时，DPC 算法没有考虑到局部数据结构；b)DPC 算法采用启发式的决策图来选取聚类中心，缺乏可靠的选择标准。例如，当样本集群中具有密度分布不均匀的情况时，DPC 算法的聚类结果往往不太理想，而具有不同密度的簇在数据集中是非常常见的。图 1(a)~(c)为 DPC 在不同输入参数下的聚类结果，与(d)中本文算法的聚类结果相比，可以看出，DPC 算法没有检测出样本数据集中所有簇，它将原本由三个簇组成的样本数据集聚类为两个簇。当遇到类似的密度不均匀数据集时，DPC 算法无法给出准确的聚类结果。

这几年来不断有学者针对 DPC 算法的不足作出改进，但同时也产生了新的问题。文献[13]基于信息熵理论提出了一种自动确定最佳输入参数的改进算法，但是该算法仍然没有解决对密度不均匀数据集聚类效果不佳的问题，并且确定参数的时间成本大大增加。文献[14]将 DPC 算法和 Chameleon 算法相结合,提出了 E_CFSFDP 算法，解决了 DPC 算法难以识别低密度簇的问题，但是其模型较为复杂。文献[15]提出了

**收稿日期：**2018-08-31；**修回日期：**2018-10-25　**基金项目：**国家自然科学基金资助项目（61273302）

**作者简介：**杨震（1994-），男，福建南平人，硕士研究生，主要研究方向为聚类分析、轨迹预测（nudt_yz1994@163.com）；王红军（1968-），男，江苏镇江人，教授，博士，主要研究方向为移动通信网、认知电子战.

DPC-KNN 算法，结合 KNN 思想重新定义局部密度，一定程度上提升了算法对密度不均匀数据集的聚类效果，但是该算法在聚类中心的选择上与 DPC 算法一样缺乏明确的标准。



(a)DPC($p$=1%) (b) DPC($p$=2%)

(c) DPC($p$=5%) (d)ADPC-WKNN($p$=2%)

图 1 DPC 算法与本文提出的 ADPC-WKNN 算法对样本数据集的聚类结果对比

Fig. 1 Comparison of clustering results between DPC algorithm and ADPC-WKNN algorithm on sample dataset

本文提出了一种基于加权 K 近邻的改进 DPC 算法（ADPC-WKNN），结合加权 K 近邻的思想重新定义并计算了局部密度，并根据样本点的权值斜率变化趋势找到聚类中心与其余点之间的临界点，解决了 DPC 算法在选取聚类中心时缺乏明确标准的不足，一定程度上避免了多选或漏选聚类中心所带来的误差。在人工数据集上的实验结果证明了本文算法的可行性。为了评估 ADPC-WKNN 算法的性能，在 UCI 数据集上对 ADPC-WKNN 算法、DPC 算法、DBSCAN 算法以及 K-MEANS++算法进行了对比实验，结果表明，在绝大多数情况下，ADPC-WKNN 算法的聚类性能更好。

## 1 算法简介

### 1.1 DPC 算法

DPC 算法的基本思想如下：聚类中心的特征在于其密度高于其周边样本，并且与具有较高密度的样本的距离相对较大。该算法使用了两个重要的量，一个是各样本点的局部密度 $\rho_i$，另一个是各样本点与具有更高局部密度样本点的最小距离 $\delta_i$。这两个量分别对应于该算法基本思想中的两个假设，即聚类中心的局部密度高于周围点的局部密度，并且与具有较高密度点的距离相对较大。接下来将详细介绍这两个量的计算方法。

假设存在一个数据集 $S=\{x_i\}_{i=1}^{N}$，$N$ 为样本点个数。首先要计算出各样本点之间的距离矩阵，$d(x_i,x_j)$表示样本点 $x_i$ 到样本点 $x_j$ 之间的欧氏距离。样本点 $x_i$ 的局部密度表示为 $\rho_i$，其计算公式如下：

$$\rho_i=\sum_j \chi(d_{ij}-d_c) \tag{1}$$

$$\chi(x)=\begin{cases}1, x<0\\ 0, x\ge 0\end{cases} \tag{2}$$

其中:$d_c$ 代表截断距离，$\rho_i$ 表示以点 $x_i$ 为圆心、$d_c$ 为半径的圆中包含的所有其余样本点的数量。

但是，对于数据量较小的数据集，式(1)有时会导致某些样本点具有同样的局部密度，从而影响聚类结果的准确性。因此，Rodriguez 和 Laio 还提供了另一种局部密度计算方法，采用高斯核函数来定义 $\rho_i$，前者可称为硬阈值，后者可称为软阈值。如下所示：

$$\rho_i=\sum_j e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \tag{3}$$

$d_c$ 是 DPC 算法中唯一的输入参数，选择 $d_c$ 的过程实际上是选择数据集中所有点的平均邻居点数量的过程。假设有数据集 $S$ 由 $N$ 个样本点组成，在求出数据集中所有样本点两两之间的距离之后，将距离值按从小到大的顺序排列，得到由 $N^2/2$ 个距离值组成的向量，$d_c$ 通常由距离总个数与用户输入的百分比 $p$ 的乘积所对应的向量中某个距离值表示。因此，实际上 DPC 算法唯一由用户输入的参数为百分比 $p$。

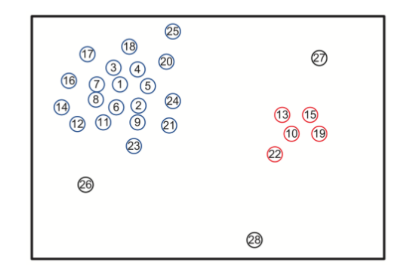
另一个量 $\delta_i$ 的计算则非常简单，它表示点 $x_i$ 与具有更高局部密度的任何其他样本点之间的最小距离，其定义如下：

$$\delta_i=\min_{j:\rho_j>\rho_i}(d_{ij}) \tag{4}$$

特别地，当样本点 $i$ 的局部密度为所有样本点中最高时，其 $\delta_i$ 的计算公式如下：

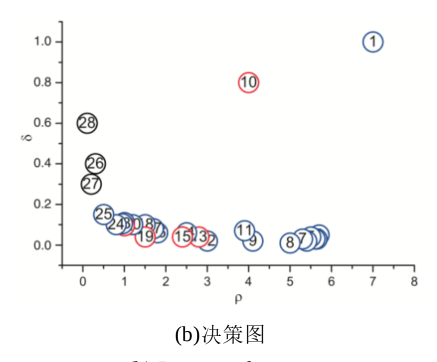
$$\delta_i=\max_j(d_{ij}) \tag{5}$$

只有同时具有较高 $\rho_i$ 和 $\delta_i$ 的样本点才被考虑为聚类中心，并且，Rodriguez 和 Laio 引入了决策图来帮助确定聚类中心，如图 2 所示，(b)中决策图的横轴表示 $\rho_i$，纵轴表示 $\delta_i$，决策图可以直观地反映出各样本点这两个量的分布情况。



(a)数据集分布

(a) Data set distribution



(b)决策图

(b) Decision diagram

图 2 数据集分布与决策图示例

Fig. 2 Data set distribution and decision diagram

DPC 算法的具体流程如下所示：

Algorithm DPC

1:**Input**: 样本点数据集 $S=\{x_i\}_{i=1}^{N}$,百分比 $p$。

2:**Output**:聚类索引的标量 $y$。

3: 计算数据集样本点两两之间的距离，得到按距离值升序排列的向量；

4:根据式(1)或(3)计算各样本点的 $\rho_i$；

5:根据式(4)计算各样本点的 $\delta_i$；

6:根据计算得到的 $\rho_i$ 与 $\delta_i$ 绘制决策图，并选取聚类中心；

7: 将其余样本点按局部密度大小分配至距离最近的更高密度点所在簇；

8:得到聚类索引的标量 $y$。

### 1.2 K 近邻算法

K 近邻算法又称为 KNN 算法，已经被广泛用于分类、回归、密度估计及模式识别等领域[16]。顾名思义，这种算法的目的就是在所有样本中找到距离目标样本最近的 K 个邻居，样本之间的距离通常由欧氏距离表示。

KNN 算法的原理比较简单，仍然假设存在数据集 $S=\{x_i\}_{i=1}^{N}$，$N$ 为样本个数。计算出目标样本 $x_i$ 与剩下 $N$-1 个样本的距离后，将距离值按升序排列，前 K 个距离值所对应的样本即距离目标样本 $x_i$ 最近的 K 个邻居，表示为 KNN($x_i$)。如图 3 所示，KNN 算法用于分类时，其基本思路为：如果一个样本在特征空间中的 K 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别，并且，其所选择的邻居都是已经正确分类的对象。

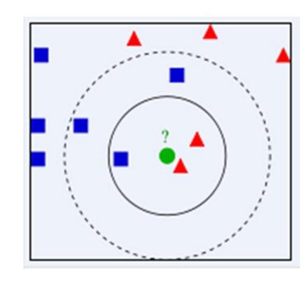



图 3 KNN 算法分类原理

Fig. 3 Classification principle of KNN algorithm

## 2 ADPC-WKNN 算法

为了提高 DPC 算法应用在密度不均匀数据集时的表现，本文基于加权 K 近邻算法的基本理念，重新定义了局部密度 $\rho_i$ 的计算方法。并且，为了克服 DPC 算法在选取聚类中心时缺乏标准的不足，本文根据样本点权值趋势给出了判别聚类中心与剩余点的明确标准，实现了聚类中心的自动选择。

DPC 算法所定义的局部密度存在对数据的局部结构不敏感的缺点，特别是数据集中不同簇的密度存在很大差异时，局部密度的变化会导致选取聚类中心时存在很大的差异。本文将加权 KNN 的概念引入到局部密度的计算中，采用反函数与高斯核函数乘积和的加权形式来表示新的局部密度。$NN_k(x_i)$表示所有样本点中与点 $x_i$ 的距离排名(由小到大)为 K 的点，KNN($x_i$)定义为

$$KNN(x_i)=\{j\in S\,|\,d(x_i,x_j)\le d(x_i,NN_k(x_i))\} \tag{6}$$

新的局部密度表示如下：

$$\rho_i=\sum_{x_j\in KNN(x_i)}\frac{1}{c+d(x_i,x_j)}*\exp(d(x_i,x_j)^{\frac{3}{2}}) \tag{7}$$

式(7)中，$c$ 表示数据集中所有样本点两两之间的距离和，唯一的参数为 K，其确定方法与截断距离 $d_c$ 类似，由用户指定一个百分比 $p$，K=$p$×$N$，$N$ 为数据集中所有样本点的数量。

图 4 为 DPC 算法、DPC-KNN 算法以及本文提出的 ADPC-WKNN 算法对 R15 数据集聚类时的决策图，从图中可以看出，(a)中 DPC 算法的决策图里聚类中心分布散乱，其中局部密度最低的聚类中心位于横轴中轴附近，其局部密度大小位于所有样本点里的中游水准，作为聚类中心的特性并不明显。(b)(c)中决策图的聚类中心更为紧凑，(c)中聚类中心的局部密度均接近于所有样本点局部密度的最大值，说明 ADPC-WKNN 算法新定义的局部密度公式能更加突出聚类中心的特性。

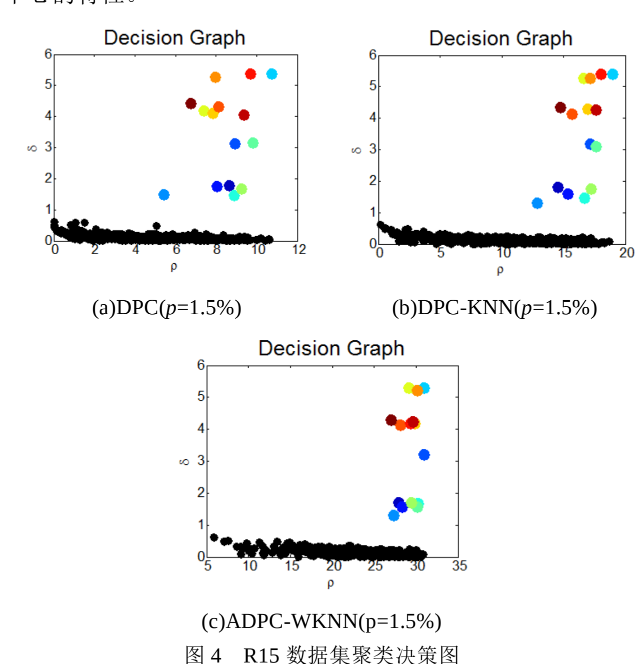



图 4 R15 数据集聚类决策图

Fig. 4 Clustering decision diagram of R15 dataset

由于 DPC 算法在选取聚类中心需要进行人工决策，使得聚类过程带有一定的主观性和随机性，难以从量化的角度确定聚类中心，不利于算法的应用。鉴于此，本文通过分析 $\rho_i$ 和 $\delta_i$ 的统计特性，提出了一种基于二者归一化乘积 $\gamma_i$ 的聚类中心临界点判别法，从而实现自动选择聚类中心的目的。该判别法的基本思想：根据 DPC 选取聚类中心的原则，通过 $\gamma_i$ 的大小差异评测样本点的特征，根据斜率变化趋势判别出聚类中心临界点，从而将临界点之前的样本点自动确定为聚类中心，将临界点及其之后的样本点根据分配原则完成聚类。

以文献[6]中使用的 GDP 数据集为例，选择 $p$=1.5%，对 $\rho_i$ 和 $\delta_i$ 进行归一化处理后求出各样本点的 $\gamma_i$，将样本点权值按降序排列并取前 40 个点，如图 5 所示，$\gamma_i$ 越大的样本点越有可能是聚类中心，样本点的权值呈先快速下降再稳定的趋势，但是下降的程度不同。因此，其中相对于初始点斜率变化趋势最大的样本点可被看做聚类中心的临界点，定义这个斜率变化趋势为 $tend_i$：

$$tend_i=(i-1)\frac{\gamma_{i+1}-\gamma_i}{\gamma_i-\gamma_1} \tag{8}$$

从式(8)可以看出，斜率变化趋势即为样本点 $i$ 到 $i$+1 斜率与样本点 $i$ 到初始点斜率的商，临界点被定义为为拥有最大 $tend_i$ 的样本点。如图 6 所示，第五个样本点拥有最大的斜率变化趋势，被判定为临界点，则将图 5 排序图中的前五个样本点选作聚类中心，完成聚类。

图 7 展示了选择不同个数聚类中心的聚类结果，可以看出，当聚类中心为五个时聚类效果最好，这也与文献[6]的聚类结果一致。

ADPC-WKNN 算法的具体流程如下：

Algorithm ADPC-WKNN

1:**Input**: 样本点数据集 $S=\{x_i\}_{i=1}^{N}$，百分比 $p$。

2:**Output**:聚类索引的标量 $y$。

3: 计算数据集样本点两两之间的距离，得到按距离值升序排列的向

量；

4:根据公式(6)计算各样本点的 $\rho_i$；

5:根据公式(4)计算各样本点的 $\delta_i$；

6:根据计算得到的 $\rho_i$ 与 $\delta_i$ 算出各样本点的 $\gamma_i$；

7:将各样本点的 $\gamma_i$ 降序排列，并计算斜率变化趋势 $tend_i$，找到最大值对应的临界点，将临界点之前的样本点选作聚类中心；

8:将其余样本点按局部密度大小分配至距离最近的更高密度点所在簇；

9:得到聚类索引的标量 $y$。

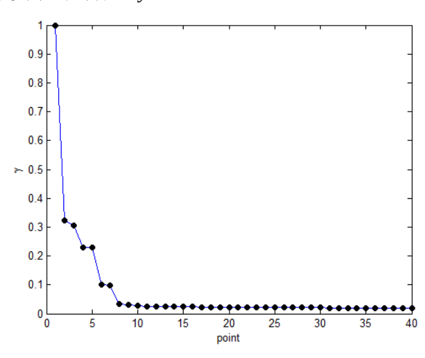



图 5　样本点权值排序图

Fig. 5　Sample point weight sorting graph

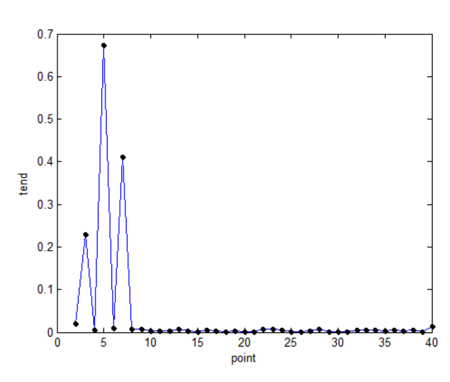
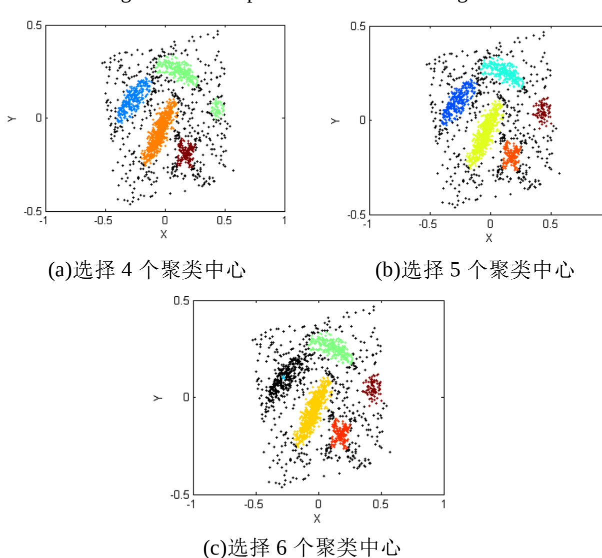



图 6　临界点判别图

Fig. 6　Critical point discrimination diagram



(a)选择 4 个聚类中心　(b)选择 5 个聚类中心

(c)选择 6 个聚类中心

图 7　选择不同个数聚类中心的聚类结果

Fig. 7　Clustering results for different numbers of cluster centers

## 3　仿真实验

为了测试 ADPC-WKNN 算法的聚类性能，本文采用 5 个经典人工数据集以及 5 个 UCI 上的真实数据集进行实验，对比算法为划分聚类经典算法 K-means++，密度聚类经典算法 DBSCAN 以及 DPC 算法。

### 3.1　实验环境与数据集

本文的实验环境为 Windows10 64 位操作系统，Intel Core i7-6700HQ @2.60 GHz CPU，8 GB 内存，采用 MATLAB2014a 进行实验。人工数据集及 UCI 真实数据集属性分别如表 1、2 所示。

表 1　人工数据集

Table 1　Artificial dataset

| 数据集 | 样本数 | 维数 | 类别数 | 来源 |
|---|---|---|---|---|
| Spiral | 312 | 2 | 3 | [17] |
| R15 | 600 | 2 | 15 | [18] |
| Flame | 240 | 2 | 2 | [19] |
| Jain | 373 | 2 | 2 | [20] |
| Aggregation | 788 | 2 | 7 | [21] |

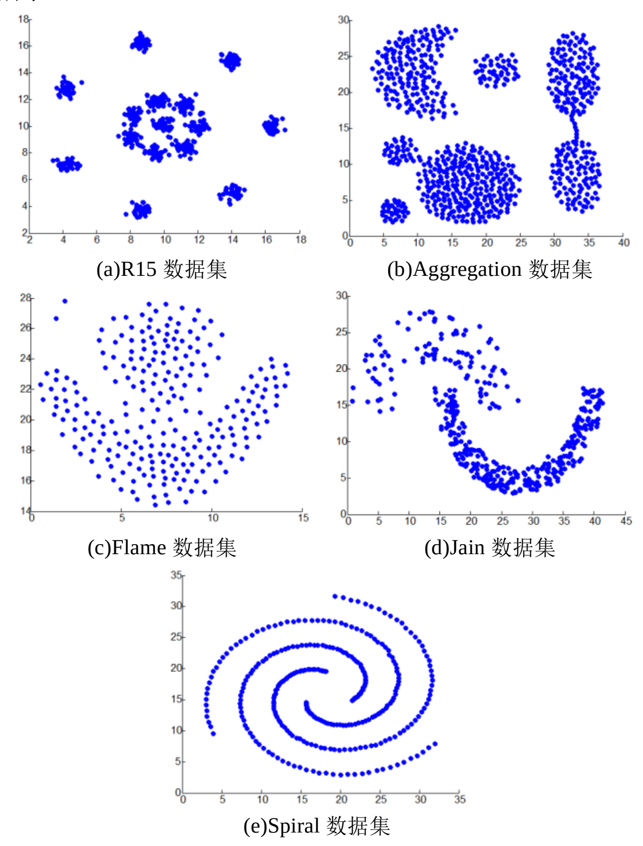
表 2　UCI 数据集

Table 2　UCI dataset

| 数据集 | 样本数 | 维数 | 类别数 | 来源 |
|---|---|---|---|---|
| Iris | 150 | 4 | 3 | [22] |
| Seeds | 210 | 7 | 3 | [22] |
| Zoo | 101 | 18 | 7 | [22] |
| Waveform | 5000 | 21 | 3 | [22] |
| Wine | 178 | 13 | 3 | [22] |

### 3.2　二维人工数据集聚类结果图及分析

本文实验中使用的人工数据集各样本点分布情况如图 8 所示。



(a)R15 数据集　(b)Aggregation 数据集

(c)Flame 数据集　(d)Jain 数据集

(e)Spiral 数据集

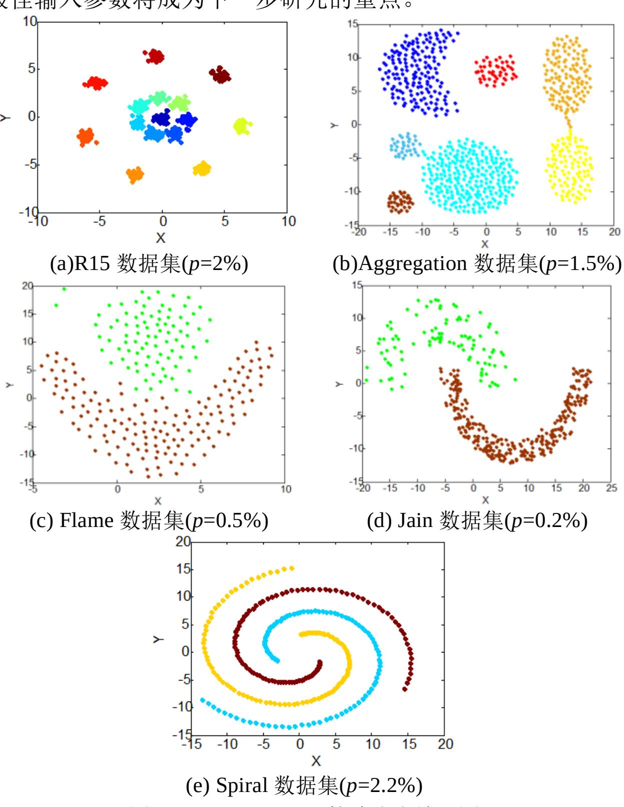
图 8　人工数据集样本点分布图例

Fig. 8　Artificial dataset sample point distribution

采用本文提出的 ADPC-WKNN 算法对这 5 个数据集进行聚类，其结果如图 9 所示。

从图中可以看出，ADPC-WKNN 在 5 个人工数据集上均取得了良好的聚类效果，但是输入参数 $p$ 的跨度较大，尤其

是 Spiral 数据集和 Jain 数据集，在样本数十分接近的情况下，输入参数分别为 2.2%和 0.2%。因此，如何辅助用户决策出最佳输入参数将成为下一步研究的重点。



(a)R15 数据集($p$=2%)　(b)Aggregation 数据集($p$=1.5%)

(c) Flame 数据集($p$=0.5%)　(d) Jain 数据集($p$=0.2%)

(e) Spiral 数据集($p$=2.2%)

图 9　ADPC-WKNN 算法聚类结果图

Fig. 9　Clustering result graph of ADPC-WKNN algorithm

### 3.3　算法聚类性能评估分析

本文采用准确率(accuracy)、召回率(recall)以及归一化互信息(normalized mutual information，NMI)来评估各算法的聚类性能，其中准确率和召回率是广泛用于信息检索和统计学领域的两个度量指标，通常用来评价聚类结果的好坏，归一化互信息则是变量之间相互依赖性的度量。三种评价指标的范围均在 0 到 1 之间，且值与聚类性能好坏成正相关。下面简要介绍这三个评价指标的计算公式。

假设 $P_j$ 为已知的人工标注过的簇，$C_j$ 为经过聚类后的簇，各指标计算公式如下：

$$Accuracy(P_j,C_i)=\frac{|P_j\cap C_i|}{|C_i|} \tag{9}$$

$$Recall(P_j,C_i)=\frac{|P_j\cap C_i|}{|P_j|} \tag{10}$$

$$NMI(P_j,C_i)=\frac{I(P_j,C_i)}{\sqrt{H(P_j)H(C_i)}} \tag{11}$$

实验结果如图 10~15、表 3~5 所示，为了更直观地展示各算法在性能指标上的差异，图中纵轴代表的数据均不是从 0 开始。

具体的实验结果如表 3~ 5 所示，表格中加粗的数字代表在此数据集上最好的结果。

从图 10~15 与表 3~5 可以看出，本文提出的 ADPC-WKNN 算法在三个聚类性能评估指标上普遍优于 DPC 算法、DBSCAN 算法与 K-MEANS++算法，对于前五个人工数据集，ADPC-WKNN 算法的三项性能指标均为最高，说明本文提出的 ADPC-WKNN 算法在二维数据集上性能优越。对于 UCI 数据集，Iris 数据集和 Seeds 数据集等维数不是特别高的数据集，本文算法在三项性能指标上仍然是最高的，但是在 Zoo、Waveform 和 Wine 等高维数据集上，ADPC-WKNN 算法的聚类效果并不突出，在 Waveform 数据集上，本文只有准确率与召回率两项指标是最高的，而在 Zoo 及 Wine 数据集上，本文算法的三项性能指标均不是最高的。由此可见，本文提出的 ADPC-WKNN 算法在中低维数据集上的聚类效果良好，其处理高维数据集的能力还有所欠缺。

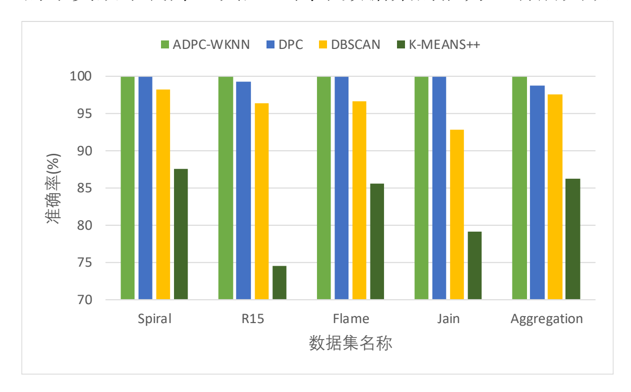



图 10　人工数据集上各算法准确率对比

Fig. 10　Accuracy comparison of algorithms on artificial data sets

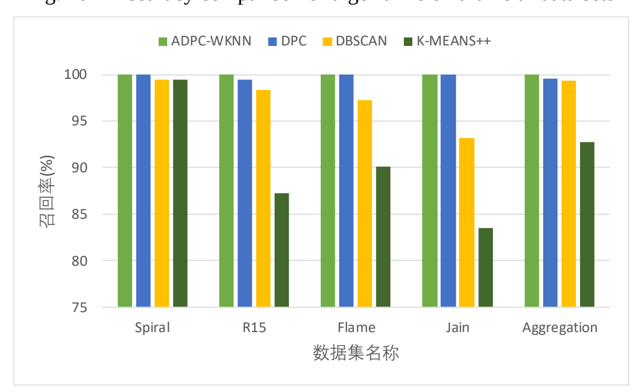



图 11　人工数据集上各算法召回率对比

Fig. 11　Recall comparison of algorithms on artificial data sets

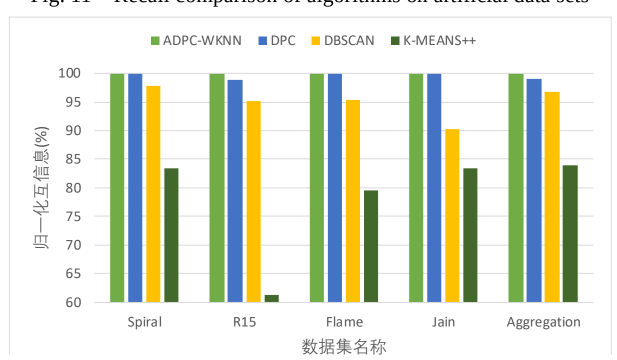



图 12　人工数据集上各算法归一化互信息对比

Fig. 12　Normalized mutual information comparison of algorithms on artificial data sets

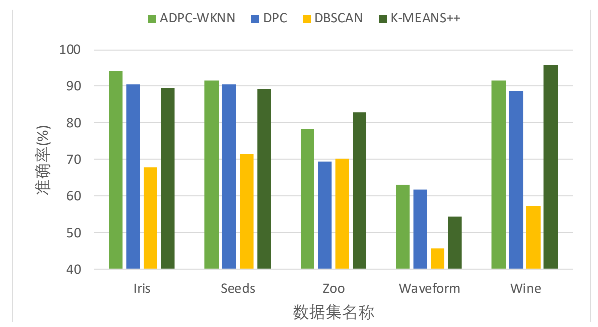



图 13　UCI 数据集上各算法准确率对比

Fig. 13　Accuracy comparison of algorithms on UCI data sets

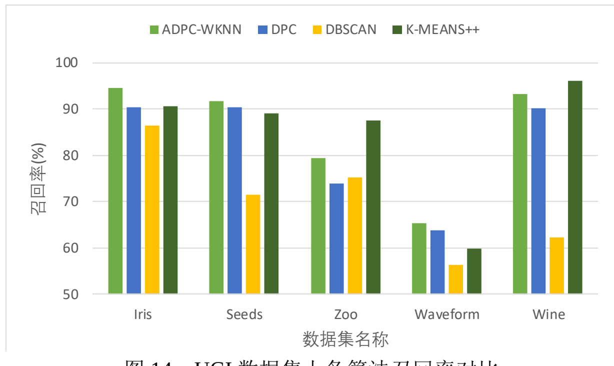



图 14 UCI 数据集上各算法召回率对比

Fig. 14 Recall comparison of algorithms on UCI data sets

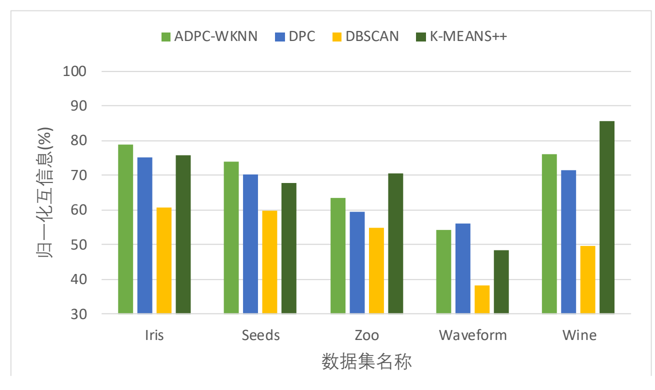



图 15 UCI 数据集上各算法归一化互信息对比

Fig. 15 Normalized mutual information comparison of algorithms on UCI data sets

表 3 各算法准确率

Table 3 Algorithm accuracy

| 数据集 | Accuracy(%) | | | |
|---|---|---|---|---|
| | ADPC-WKNN | DPC | DBSCAN | K-MEANS++ |
| Spiral | **100** | **100** | 98.3 | 87.6 |
| R15 | **100** | 99.3 | 96.4 | 74.5 |
| Flame | **100** | **100** | 96.7 | 85.6 |
| Jain | **100** | **100** | 92.8 | 79.1 |
| Aggregation | **100** | 98.8 | 97.6 | 86.2 |
| Iris | **94.2** | 90.6 | 67.8 | 89.5 |
| Seeds | **91.7** | 90.4 | 71.4 | 89.1 |
| Zoo | 78.5 | 69.3 | 70.1 | **82.9** |
| Waveform | **63.1** | 61.8 | 45.7 | 54.3 |
| Wine | 91.7 | 88.6 | 57.3 | **95.8** |

表 4 各算法召回率

Table 4 Algorithm recall

| 数据集 | Recall(%) | | | |
|---|---|---|---|---|
| | ADPC-WKNN | DPC | DBSCAN | K-MEANS++ |
| Spiral | **100** | **100** | 99.5 | 99.4 |
| R15 | **100** | 99.5 | 98.4 | 87.2 |
| Flame | **100** | **100** | 97.3 | 90.1 |
| Jain | **100** | **100** | 93.2 | 83.5 |
| Aggregation | **100** | 99.6 | 99.3 | 92.7 |
| Iris | **94.6** | 90.3 | 86.5 | 90.7 |
| Seeds | **91.7** | 90.4 | 71.4 | 89.1 |
| Zoo | 79.3 | 73.8 | 75.2 | **87.6** |
| Waveform | **65.3** | 63.7 | 56.4 | 59.8 |
| Wine | 93.2 | 90.1 | 62.3 | **96.2** |

表 5 各算法归一化互信息

Table 3 Algorithm normalized mutual information

| 数据集 | NMI(%) | | | |
|---|---|---|---|---|
| | ADPC-WKNN | DPC | DBSCAN | K-MEANS++ |
| Spiral | **100** | **100** | 97.8 | 83.4 |
| R15 | **100** | 98.9 | 95.2 | 61.3 |
| Flame | **100** | **100** | 95.4 | 79.6 |
| Jain | **100** | **100** | 90.3 | 83.4 |
| Aggregation | **100** | 99.1 | 96.8 | 83.9 |
| Iris | **78.8** | 75.3 | 60.7 | 75.8 |
| Seeds | **73.9** | 70.3 | 59.9 | 67.8 |
| Zoo | 63.5 | 59.3 | 54.7 | **70.4** |
| Waveform | 54.3 | **56.1** | 38.2 | 48.3 |
| Wine | 76.1 | 71.4 | 49.5 | **85.7** |

### 3.4 算法复杂度分析

设 $N$ 为数据集点个数，DPC 算法的时间复杂度为 $O(N^2)$，其复杂度主要来源于计算 $N$ 个数据点两两之间的距离，相比于 DPC 算法，本文提出的 ADPC-WKNN 算法由于加入了聚类中心的自动选择，需要多计算一个量 $\gamma_i$，故 ADPC-WKNN 算法的时间复杂度为 $O(N^2)+O(N)\sim O(N^2)$。DBSCAN 算法与 K-MEANS++算法的时间复杂度分别为 $O(N^2)$与 $O(N)$，这是因为 K-MEANS++算法作为划分聚类算法，无须考虑数据点两两之间的距离，只需考虑各数据点与指定的数个聚类中心的距离。但是，K-MEANS++算法需要人为指定聚类中心的个数，且对于形状不规则的数据集聚类效果不好。

## 4 结束语

DPC 算法在对密度分布不均匀的数据集聚类时效果不理想，且选取聚类中心时缺乏明确的标准。本文针对这两个问题提出了一种基于加权 K 近邻的改进密度峰值聚类算法：ADPC-WKNN，该算法结合加权 K 近邻的思想，采用反函数与高斯核函数的乘积形式重新定义了局部密度，并且基于权值斜率变化趋势来确定聚类中心临界点，有效解决了原算法存在的问题。在人工数据集及 UCI 真实数据集上的实验结果表明，ADPC-WKNN 算法能准确识别出二维数据集的所有簇，且在聚类性能指标上也普遍高于原算法以及经典密度聚类算法 DBSCAN 与经典划分算法 KMEANS++。在今后的研究过程中，如何确定算法的最佳输入参数与提高算法在高维数据集上的聚类性能将是下一步的研究重点。

## 参考文献：

[1] Jain A K, Murty M N, Flynn P J. Data clustering: a review [J]. Acm Computing Surveys, 1999, 31(3): 264-323.

[2] Han Jiawei. Data mining: concepts and techniques [M].San Francisco: Morgan Kaufmann Publishers Inc., 2005: 65-67.

[3] Arthur D, Vassilvitskii S. K-means+: the advantages of careful seeding [C]//Proc of the 18th ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana. Society for Industrial and Applied Mathematics.New York:ACM Press, 2007: 1027-1035.

[4] Ester M, Kriegel H P, Xu Xiaowei. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise [C]//Proc of International Conference on Knowledge Discovery and Data Mining.Pola Alto:AAAI Press, 1996: 226-231.

[5] Hou Jian, Gao Huijun, Li Xuelong. DSets-DBSCAN: a parameter-free clustering algorithm [J]. IEEE Trans on Image Processing, 2016, 25 (7): 3182-3193.

[6] Rodriguez A, Laio A. Machine learning: clustering by fast search and find of density peaks. [J]. Science, 2014, 344 (6191): 1492.

[7] Sun Kang, Geng Xiurui, Ji Luyan. Exemplar component analysis: a fast band selection method for hyperspectral imagery [J]. IEEE Geoscience & Remote Sensing Letters, 2015, 12(5): 998-1002.

[8] Wang Baoyan, Zhang Jian, Liu Yi, *et al*. Clustering sentences with density peaks for multi-document summarization [C]//Proc of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015: 1262-1267.

[9] Xie Ke, Wu Jin, Yang Wankou, *et al*. K-Means Clustering Based on Density for Scene Image Classification [C]// Proc of Chinese Intelligent Automation Conference. 2015: 379-386.

[10] Chen Yewang, Lai Dehe, Han Qi, *et al*. A new method to estimate ages of facial image for large database [J]. Multimedia Tools & Applications, 2016, 75(5): 2877-2895.

[11] Chen Guijun, Zhang Xueying, Wang Zizhong, *et al*. Robust support vector data description for outlier detection with noise or uncertain data [J]. Knowledge-Based Systems, 2015, 90(C): 129-137.

[12] Shamshirband S, Amini A, Anuar N B, *et al*. D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks [J]. Measurement, 2014, 55(9): 212-226.

[13] Wang Shuliang, Wang Dakui, Li Caoyuan, *et al*. Clustering by fast search and find of density peaks with data field [J]. Computer Science, 2016, 25(3): 397-402.

[14] Zhang Wenkai, Li Jing. Extended fast search clustering algorithm: widely density clusters, no density peaks [J]. Computer Science, 2015, 5(7): 415-423.

[15] Du Mingjing, Ding Shifei, Jia Hongjie.Study on density peaks clustering based on K-nearest neighbors and principal component analysis [J]. Knowledge-Based Systems, 2016, 99: 135-145.

[16] Chen Huiling, Yang Bo, Wang Gang, *et al*. A novel bankruptcy prediction model based on an adaptive fuzzy-nearest neighbor method [J]. Knowledge-Based Systems, 2011, 24: 1348-1359.

[17] Chang Hong, Yeung D Y. Robust path-based spectral clustering [M]. Elsevier Science Inc. 2008: 17-23.

[18] Veenman C J, Reinders M J T, Backer E. A maximum variance cluster algorithm [J]. Pattern Analysis & Machine Intelligence IEEE Trans on, 2002, 24(9): 1273-1280.

[19] Fu Limin, Medico E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data[J].BMC Bioinformatics, 2007, 8(1): 3.

[20] Sarat C, Zhu, Yongfang, Jain A K, *et al*. Statistical models for assessing the individuality of fingerprints [J]. IEEE Trans on Information Forensics & Security, 2007, 2(3): 391-401.

[21] Gionis A, Mannila H, Tsaparas P. Clustering aggregation [J]. ACM Trans on Knowledge Discovery from Data, 2007, 1(1): 4.

[22] Bache K, Lichman M. UCI machine learning repository [EB/OL]. (2017-11-30). http://archive. ics. uci. edu/ml.